PHILIPS ADDS NEW DIMENSIONS TO INFORMATION MANAGEMENT

P9070

PC INTERFACE

Administrator's Guide

PHILIPS

A publication of:

Philips Telecommunicatie en Data Systemen Nederland B.V.
SSS - Training & Documentation
P.O. Box 245
7300 AE Apeldoorn, The Netherlands

Printed in The Netherlands, March 1987

Order number: 5122 995 11901

Manual number: G316A

PC-Interface Administrator's Guide

UNIX System V
Motorola 8000 Computers

87601909A
September 1986

Version 2.7

Motorola Computer Systems, Inc.

You are responsible for observing the licensing restrictions associated with DOS applications you install on the virtual drive. Unless the licensing agreement permits you to share the software, you should only install applications in your own directories, and you should remove execute permission for all other users from those directories.

Public-domain software and DOS applications for which you have a multiuser licensing agreement can be installed in public bin directories on the virtual drive. Generally, the system administrator installs these packages and can tell you what applications are available at your installation and where they are. (We recommend that each application be installed in a separate subdirectory of \ USR \ PCI \ PCBIN.)

Be sure your search path specifies these directories so DOS can find the applications.

Motorola, Inc.
10700 North De Anza Boulevard
Cupertino, California 95014

About This Book

This manual explains how to install and administer the Ethernet and RS-232 versions of the PC-Interface software on a Motorola 8000 host running the UNIX System V operating system. It thoroughly covers the system administrator's responsibilities for installation and day-to-day operation and maintenance of the PC-Interface software. For information on the operation of PC-Interface from the user's point of view, refer to the *PC-Interface User's Guide.*

The PC-Interfaces software allows you to:

- Integrate your personal computer into your existing UNIX host computer network.
- Use the file services of a UNIX host while running DOS applications as if the host were a colocated fixed disk.
- Conduct a standard UNIX session from your personal computer.

The Motorola 8000 host can support a network of personal computers including any of the IBM Personal Computer line (PC, XT, and AT), and IBM-compatible personal computers manufactured by other vendors.

Throughout this manual, the term "personal computer" refers to any of the personal computers listed above. The term "host" refers to the Motorola 8000 computers. The term "DOS" refers to the Microsoft Disk Operating System program (Version 2.0 or later).

Who Should Use This Manual

This manual is for the PC-Interface system administrator—the person who installs the host software and manages the day-to-day operation of the system.

This guide assumes that you are familiar with UNIX commands and basic system administrator activites such as loading a tape or mounting a diskette, making directories, adding users, and booting the 8000 computer. It also assumes that you are familiar with the information contained in the *PC-Interface User's Guide*. Familiarity with basic DOS commands and with your network hardware is useful but not required.

Typographic Conventions

This guide uses several typographic conventions to help you distinguish between DOS and UNIX commands and file names and to help you recognize text that you must type exactly as shown. These conventions are:

- References to UNIX commands and file names are in lowercase bold (for example, **lpadmin**). DOS commands and file names are in uppercase (for example, DIR D:/U/BOB).

- Examples showing exactly what you type use the lowercase Courier font for both UNIX and DOS commands:

  uren c:/u/mary/messagetoall c:/u/mary/message

- Italics indicate generic information for which you should substitute actual values for your system. For example, the following means you should supply the names of the source and target files to be used by the UREN command:

  uren **file1** **file2**

- The path separator and switch character for the operating system under discussion are used: the slash (/) and hyphen (-) for UNIX, and the backslash (\ ) **and slash (/)** for DOS.

- Prompts are shown (in bold) only in examples that might be confusing without them.

- Examples do not explicitly show carriage returns. It is assumed that you type a carriage return at the end of each line.

Organization of this Manual

This manual has six chapters and one appendix as follows:

Chapter 1. Introduction acquaints you with the PC-Interface software and configuration requirements for your UNIX host computer.

Chapter 2. About PC-Interface describes the operation of the PC-Interface software in more depth and discusses aspects of the DOS environment with which the administrator may not be familiar.

Chapter 3. PC-Interface Networks discusses the networks that support PC-Interface operation.

Chapter 4. PC-Interface Administration tells you how to get started with the administration of PC-Interface, how to support day-to-day operation of PC-Interface, and how to tailor your host system to better suit the way your system is used.

Chapter 5. The PC-Interface System Architecture is a thorough description of the system architecture for the experienced administrator who wants to know more about PC-Interface operation.

Chapter 6. Some Problem-Solving Tools for PC-Interface suggests some trouble-shooting procedures for you to follow when PC-Interface users experience problems they can't solve themselves.

Appendix A. PC-Interface System Messages explains all the messages you might receive from PC-Interface, including installation and removal messages.

Related Publications

Other PC-Interface publications include:

- ***PC-Interface User's Guide,*** Locus Computing Corporation.

- **Functional Specification:** *Virtual RS-232 Communications Handler for PC-Interface,* Locus Computing Corporation.
- *PC-Interface Network Driver Specification,* Locus Computing Corporation.
- *Transparent Integration of UNIX and MS-DOS,* Locus Computing Corporation.

Host hardware publications include:

UNIX Publications include:

- *8000 PC Interface Software Release Guide*
- *8000 UNIX System V User Reference Manual.*
- *8000 UNIX System V Adminstrator Reference Manual.*
- *8000 UNIX System V Programmer Reference Manual.*
- *8000 UNIX System V User Guide and Essential Utilities Manual.*
- *8000 OfficeLAN Administrator's Guide*

Personal computer publications include:

- *Guide to Operations.*
- *Disk Operating System.*

Contents

Chapter 1. Introduction

Chapter 2. About PC-Interface

Chapter 3. PC-Interface Networks

Chapter 4. PC-Interface Administration

Chapter 5. PC-Interface System Architecture

Chapter 6. Some Problem-Solving Tools for PC-Interface

Appendix A. PC-Interface Messages

Chapter 1. Introduction

PC-Interface

The PC-Interface software package allows you to bridge the gap between your personal computer and UNIX hosts. With PC-Interface and either an RS-232 or an Ethernet connection to a UNIX host, you can run DOS applications using data from the UNIX file system. You can store files and even DOS applications on the UNIX host and access them just as if they were on a local fixed disk. By taking advantage of the increased storage capacity of the UNIX file system, your personal computer without a fixed disk can operate as efficiently as a personal computer with a fixed disk.

When you use the host file services of PC-Interface, the file security capabilities of UNIX allow you to share files throughout a personal computer network while protecting your files from unauthorized access. You can also take advantage of shared host resources, such as a laser printer.

PC-Interface can also emulate a VT100 terminal, allowing you to conduct a regular UNIX session. Because PC-Interface's emulator blocks data for input and output efficiently, screen refreshing activities can be faster and load on the host can be less than running the same task on a conventional terminal.

For all its power, PC-Interface is remarkably simple to use. DOS users need to know nothing about UNIX to use host file services. Just treat the UNIX system as an enhanced disk drive connected directly to your personal computer. Users familiar with DOS and UNIX can combine host file services and terminal emulation, toggling back and forth between the two modes. For example, you could create a text file in host file services mode using a DOS word-processing package, then switch to terminal emulation mode to include that file in a UNIX mail message.

Minimum Requirements

To use the PC-Interface software over Ethernet, you must have the following:

Host and Network Requirements for Ethernet-Based Communication

- One or more Motorola 8000 computers.
- The UNIX System V Operating system.
- OfficeLAN hardware and software for each host.
- Enough Ethernet cable to connect the computers.
- Appropriate adapters to connect the Ethernet board, the Ethernet cable, and the transceiver (if used) together.
- Terminating resistors (or "terminators") for any unused connections on the network.

Host Requirements for RS-232-Based Communication

To use the PC-Interface software over RS-232, you must have the following:

- One or more Motorola 8000 computers.
- The UNIX System V operating system.
- One asynchronous communications adapter for ***each*** PC-Interface RS-232 user.

- Standard RS-232 cable to connect each asynchronous communications adapter in the host to the asynchronous communications adapter in a personal computer. The connectors used to attach the cable to the asynchronous communications adapters depend on the specific asynchronous communications adapter you use.

Personal Computer Requirements

- One or more IBM personal computers (PC, XT, or AT) or IBM-compatible personal computers configured as described in Chapter 1 of the ***PC-Interface User's Guide.***

- The PC-Interface personal computer software, contained on the PC-Interface distribution diskette.

- PC-DOS or MS-DOS, Version 2.0 or later.

Installation of Adapters in Host

The Ethernet board, required for PC-Interface Ethernet service, should be installed in your 8000 host according to the instructions in the product manual for the board. Asynchronous communications boards for RS-232 service should be installed according to the instructions in the product manual for the board.

Installing PC-Interface

The PC-Interface host-side distribution diskette or cassette tape contains all the files required for your host to support PC-Interface. It also contains an installation program that simplifies the installation process. Before reading further, you should follow the instructions in the ***S8000 PC Interface Software Release Guide (SRG).*** Once PC-Interface is installed on your host, it is automatically initialized whenever you boot the host system.

This guide assumes that you have installed PC-Interface exactly as described in the S8000 PC Interface Software Release Guide (SRG).

Chapter 2. About PC-Interface

Introduction

This chapter discusses the operation of PC-Interface and describes the two major PC-Interface services as well as the PC-Interface utilities that enhance the interface between DOS and UNIX. The discussion here is introductory. You should also be thoroughly familiar with the description of PC-Interface in the ***PC-Interface User's Guide*** to effectively administer the PC-Interface system. Some comments on the DOS environment are included at the end of this chapter for administrators who are not familiar with DOS.

Host/Personal Computer (UNIX/DOS) Integration

When the network hardware and software are installed, users can easily access both host (UNIX) and personal computer (DOS) resources. PC-Interface supplies two major services that integrate network hosts and personal computers, as well as a number of special-purpose utilities that enhance the UNIX/DOS interface. These services and utilities are described briefly below.

Host File Service

When you log into PC-Interface from your personal computer, you have access to the same UNIX files as you do when you access the UNIX host in any other way. These files are available to DOS on a virtual drive that is specified by PC-Interface in the login connection message. From the user's point of view, the virtual drive has the characteristics of a DOS drive attached to the user's personal computer. With PC-Interface the user can:

- Create or access files on the virtual drive using DOS commands or applications by specifying the virtual drive identifier as part of the file name.

- Change the default drive to the virtual drive.
- Copy files from diskette or local fixed disk to the virtual drive following the same procedures used to copy to a local disk.
- Create directories on the virtual drive using the DOS MKDIR command.
- Set DOS search paths that specify directories on the virtual drive.

In short, the user can do almost anything with the remote UNIX drive during a file service session that he can do with a local drive on a stand-alone personal computer.

Terminal Emulation Service

The PC-Interface software enables a personal computer to emulate a standard VT100 terminal. Personal computer users can therefore connect the UNIX host and conduct standard interactive UNIX sessions as though they were using terminals rather than personal computers running DOS. The advantages of using terminal emulation are 1) users more familiar with UNIX than with DOS can work in the environment with which they are comfortable, and 2) all users have access to UNIX commands that are not directly accessible from the DOS environment.

Users can freely intermix file service and terminal emulation sessions. To switch from one service to the other, you need not terminate your current work. PC-Interface allows you to suspend your work with either service at any time and switch to the other service. Unless you explicitly terminate your current service, you will find any ongoing jobs in the state you left them when you return to that service.

PC-Interface Utilities

PC-Interface provides several utility programs that enhance the user interface between DOS and UNIX, as listed below. All of them are supplied on the PC-Interface personal computer distribution diskette and can be invoked from the DOS environment of the personal computer. UNIX2DOS and DOS2UNIX are also supplied in versions that run under UNIX.

- DOS2UNIX and UNIX2DOS convert text files from DOS text format to UNIX text format and visa versa.
- PRINTER directs DOS printing to a local (personal computer) printer or the remote (UNIX) printer.
- UCHMOD changes the permission mode of files on the UNIX drive.
- UDIR lists files and directories on the virtual drive in UNIX format.
- UREN allows the user to change the names of UNIX files that have DOS-incompatible names.
- UEXEC causes UNIX programs to be executed from the DOS environment.

For information on PC-Interface utilities of special interest to the system administrator, refer to Chapter 4.

The DOS Environment

As system administrator, you should be familiar with the UNIX environment-including commonly used commands, files, messages, and procedures. Some familiarity with the DOS environment is also helpful. Following are brief definitions of some common DOS terms used in this guide. For more information on the DOS environment, refer to the related publications listed in the preface.

Application software: Programs designed for specific purposes, such as accounting or word processing, and generally commercially distributed.

Disk: Loosely, a magnetic disk. In this guide, the term ***disk*** specifically refers to a fixed disk, as opposed to a diskette. (See also ***fixed disk***.)

Diskette: A thin, flexible magnetic disk permanetly enclosed in a semirigid protective jacket. Synonymous with flexible disk and floppy disk.

Distribution diskette: The diskette packaged with the ***PC-Interface User's Guide*** containing the PC-Interface personal computer software. (PC-Interface ***host*** software is distributed on either a distribution diskette or a distribution tape—depending on the particular host system—and is packaged with the ***PC-Interface Administrator's Guide***.) (See also ***working diskette***.)

DOS drive: The personal computer hardware associated with DOS diskettes or fixed disks. DOS drives are named with letters followed by a colon. By convention, diskette drives are usually designated A: and B:, and a local fixed disk is designated C:. The contents of the disk or diskette on each drive are presented to the DOS user as separate file systems (each having its own root), unlike UNIX, which presents the contents of mounted drives as part of a single, integrated file system. (See also ***DOS volume*** and ***virtual drive***.)

DOS executable files: DOS executable files are invoked by name, like UNIX executable files. DOS, however, requires executable files to end with one of the three-letter extensions .BAT, .COM, or .EXE. (For example, executable files familiar to DOS users include AUTOEXEC.BAT and COMMAND.COM.) .BAT files are equivalent to UNIX shell files—they are text files containing a list of DOS commands that are executed when the file is invoked. AUTOEXEC.BAT is a special file contained in the root directory of the fixed disk or PC-Interface working diskette that is executed automatically whenever the personal computer is booted. .COM and .EXE files are machine-readable DOS programs that DOS recognizes and distinguishes on the basis of the file name extension (.COM or .EXE) and internal format.

DOS search paths: The directories searched by DOS for commands. The DOS search path is analogous to the UNIX search path, except:

- Under DOS, the current directory is ***always*** searched first.
- DOS search paths can include different DOS drives, each containing an independent file system.

DOS volume: The contents of a DOS disk or diskette. DOS allows you to identify the contents of a disk or diskette with a volume label (see the DOS FORMAT and VOL commands). ***Volume*** is sometimes used interchangeably with ***drive***, but ***drive*** more precisely refers to the personal computer hardware associated with a disk or diskette. (See also ***DOS drive***.)

Emulate: To simulate one system with another, so that the simulating computer system accepts the same data, executes the same programs, and achieves the same results. The PC-Interface system includes terminal emulation software that allows a personal computer to emulate a VT100 terminal.

Fixed disk: A rigid magnetic disk housed in either the system unit or an expansion unit of a personal computer, used for mass storage.

Function keys: Special keys on the personal computer keyboard that perform tasks normally requiring more than one keystroke or tasks that cannot be performed with keys representing standard characters. The function keys are labelled F1 through F10.

Virtual drive: The UNIX host when accessed from the personal computer for host file services. The PC-Interface LOGIN program identifies this drive (typically C: or D:) when the user logs in to PC-Interface. The drive is called ***virtual*** because it doesn't use the hardware of a local DOS drive, but can nevertheless be treated as one. (See also ***DOS drive***.)

Working diskette: The disk or diskette from which DOS is booted and from which PC-Interface software is loaded. The PC-Interface user creates the working diskette by combining the DOS system files with the contents of the PC-Interface distribution diskette. (See also ***distribution diskette***.)

Chapter 3. PC-Interface Networks

Introduction

This chapter describes the functional characteristics of the Ethernet and RS-232 hardware and software as they relate to the PC-Interface system. For further general information on network administration, consult appropriate Ethernet and host system hardware documentation. Network administration required specifically for PC-Interface is covered in Chapter 4.

Ethernet Characteristics

Ethernet is a high-bandwidth, low-delay communication medium and link-level data transmission protocol that allows a computer to send packets of data to other computers. The maximum bandwidth of Ethernet is 10 megabits per second. This bandwidth is shared by all computers using the network.

The PC-Interface software uses Ethernet, when selected, for all data traffic between the personal computer and the host system.

For use with terminal emulation, the PC-Interface software creates special virtual serial lines over the Ethernet between the personal computers and the host. These virtual serial lines operate independently of and concurrently with the PC-Interface file-sharing mechanism. The virtual serial lines can also be used by any personal computer software that works with serial lines to communicate with a UNIX terminal (tty) port.

UNIX Host Ethernet Hardware and Software

The UNIX host Ethernet controller (the Ethernet **hardware**) gives the UNIX host the ability to communicate over Ethernet. The controller is a board that plugs into the UNIX host backplane and is accessible under kernel control using the UNIX host Ethernet driver. The board is connected via transceiver cable to an external transceiver, which connects to the actual Ethernet cable.

The UNIX host Ethernet driver provides the interface between the Ethernet hardware and the various processes that use the Ethernet. It can handle a fixed number of **ports**, or logical connections. Each port has a set of connections. Each port has a set of **input buffers** associated with it, where incoming packets are temporarily placed until a process asks for them. Each port can be independently opened and have reads and writes performed on it. The driver is responsible for demultiplexing incoming Ethernet packets and placing the packets into input buffers reserved for the various ports based on a data field in the packet or on the Ethernet address.

Ethernet can support multiple high-level protocols simultaneously.

Personal Computer Ethernet Hardware and Software

The Ethernet controller used by PC-Interface personal computers is the EtherLink board manufactured by 3Com Corporation. The board plugs into the personal computer backplane according to instructions contained in the ***PC-Interface User's Guide***. It is accessible from the PC-Interface software via the PC-Interface Ethernet driver.

The PC-Interface Ethernet drive is the interface between the Ethernet hardware and the rest of the PC-Interface software. The driver is interrupt-driven and tuned for maximum network performance. It performs a demultiplexing function for the PC-Interface software. allowing flexibility of operation and greater system performance.

RS-232 Characteristics

RS-232 is a low-bandwidth, point-to-point communication medium and data transmission protocol that supports data transfer between two computer systems. Typical RS-232 bandwidths are under 10 kilobits per second.

PC-Interface uses the host system's terminal ports in the same manner as a normal terminal would. No physical RS-232 lines need be reserved for PC-Interface service. While file service is being provided, PC-Interface uses a simple frame-based communications protocol with moderate-sized packets. For the sake of performance, whenever terminal emulator traffic is encountered, the protocol is switched to a much lower overhead method, so that single-character transmissions do not require sending a large number of bytes. Users can arbitrarily intermix terminal emulation and file service messages.

Multiple Hosts and Networks

The PC-Interface host software can support multiple simultaneous personal computer users. Each concurrent personal computer user is handled independently of all others. The only limits are those imposed by the operating system, such as the number of processes in the system and the number of simultaneous network connections allowed.

The PC-Interface host software can also simultaneously support both Ethernet and RS-232 connections with personal computers. Because the host software is designed to handle Ethernet and RS-232 independently, your PC-Interface system can use Ethernet only, RS-232 only, or both media simultaneously.

PC-Interface networks using Ethernet can have multiple UNIX hosts as well as multiple personal computers. PC-Interface personal computer users can access any host on the network for which they have a UNIX account.

The PC-Interface personal computer user can connect to only one host at a time. If the personal computer is connected to a network that includes more than one host, the user can change form one host to another simply by logging out and logging in to another host.

The PC-Interface personal computer software supports both RS-232 and Ethernet connections. If the personal computer has the necessary hardware for both media, the user can choose either medium for a PC-Interface session.

Chapter 4. PC-Interface Administration

Introduction

This chapter discusses managing your PC-Interface system. The topics are organized in three sections, as follows:

- **Getting Started with PC-Interface:** starting and stopping host servers, and adding and removing PC-Interface users.

- **Supporting PC-Interface Operation: /etc/termcap** entries for terminal emulation, administering network printers, managing PC-Interface UNIX processes, backing up system data, copy protection, and error messages.

- **Tailoring Your PC-Interface System:** managing UNIX disk resources and load, establishing shared personal computer software libraries, changing default file permission modes, enabling and disabling terminal emulation ports, and DOS software compatibility issues.

To support everyday operation of the PC-Interface system, you must be familiar with the topics discussed in the first two sections — "Getting Started with PC-Interface" and "Supporting PC-Interface Operation." The final section, "Tailoring Your PC-Interface system," tells you how to customize and configure your host to suit the needs of your installation.

Getting Started With PC-Interface

Starting And Stopping Host Servers (Daemons)

Once the host software is installed according to the instructions in the S8000 PC Interface Software Release Guide (SRG), PC-Interface is operational each time you boot the host system. Occasionally, you may need to stop PC-Interface operation — for example, when you want to change the default **umask** by modifying the **pcistart** script. You can stop PC-Interface without shutting down the entire system. To resume PC-Interface service, you can either reboot the host system or use a special utility for restarting the PC-Interface daemons.

The two utilities **/usr/pci/bin/pcistart** and **/usr/pci/bin/pcistop** start and stop the two PC-Interface Ethernet daemons **/usr/pci/pcimapsvr.eth** and **/usr/pci/pciconsvr.eth.** (For a more complete explanation of these utilities, refer to Chapter 5.) Whenever you intend to stop PC-Interface service, you should inform all PC-Interface users who are currently logged in that service is about to be terminated, and then invoke **/usr/pci/bin/pcistop.** The **pcistop** utility does not stop PC-Interface RS-232 processes. To stop them, PC-Interface RS-232 users should log out. If necessary, you can kill PC-Interface RS-232 processes manually by running the UNIX command to find out the process ID (pid) of any currently executing PC-Interface processes. Look for any processes in the **ps** list that start with ***pci*** and kill the relevant pid. To resume PC-Interface service, either reboot or invoke **/usr/pci/bin/pcistart.**

Adding PC-Interface Users

Adding a PC-Interface user to the network is as simple as creating a UNIX account for the user and attaching the user's personal computer to the host via Ethernet or RS-232. Create the UNIX account in the usual way: by making an entry in **/etc/passwd** and creating a home directory for the user. The user's personal computer must have a Ethernet board or asynchronous communications adapter installed and connected to the Ethernet or to RS-232 cable, respectively.

To use PC-Interface, each user must have installed PC-Interface using the PC-Interface distribution diskette. (Note that no two users may simultaneously use working disks created from the same PC-Interface distribution diskette. See "Copy Protection," later in this chapter.)

When you add users, you may also need to enable additional terminal emulation ports.

Removing PC-Interface Users

You can remove a PC-Interface user from the network in any of the following ways:

- Deleting the user's UNIX account following the usual UNIX procedure.
- Disconnect the user's personal computer from the network.
- Collect the user's PC-Interface working diskette (and also the PC-Interface distribution diskette, if he has it).

Once the user's account has been removed, you should perform the same functions you would when removing any UNIX account. This would typically include deleting, archiving, or transferring files.

When you remove users, you may also want to disable terminal emulation ports. Refer to "Enabling and Disabling Terminal Emulation Ports" later in this chapter.

Supporting PC-Interface Operation

Entries in /etc/termcap for Terminal Emulation

The PC-Interface terminal emulator (EM.COM) requires no special entry in **/etc/termcap**. Since the emulator simulates a standard VT100 terminal, use the same **/etc/termcap** entry you use for a VT100 terminal. The UNIX system is preconfigured with the correct **/etc/termcap** entry.

Administering Network Printers

When the user sets PRINTER REMOTE, PC-Interface automatically spools print jobs from the personal computer to the default system printer. (Multiple remote printers are not currently supported by PC-Interface.) The only responsibility the PC-Interface system administrator has in connection with printers is to change the default system printer when necessary. Note that making such a change affects all UNIX system users, not just those connected through PC-Interface.

To change the default system printer, use the **lpadmin** command. (You must be superuser or log in as root to use **lpadmin.**) For example, to make the printer named **laser1** the default printer, type:

```
/usr/lib/lpadmin -dlaser1
```

For further information on administering UNIX printers, refer to your host system documentation.

Managing PC-Interface UNIX Processes

All PC-Interface users who establish terminal emulation or file service sessions with the UNIX host cause UNIX processes to be executed on the host. When UNIX commands are invoked during a terminal emulation session, the host executes these commands just as though they were issued from a terminal connected directly to the host. Managing these processes is not different from managing any other processes invoked from a UNIX terminal.

When file service users execute DOS commands on the virtual drive, the PC-Interface software translates the commands and executes them on the UNIX system. In general, the PC-Interface software manages these UNIX processes transparently to the user, and they rarely require intervention by the system administrator.

With the PC-Interface UEXEC command, however, PC-Interface file service users can invoke UNIX processes that are not under the control of the PC-Interface software. A user who is unfamiliar with UNIX might use UEXEC inappropriately and start a runaway process on the UNIX system — a defective process that consumes UNIX resources and never terminates. If the user cannot terminate the process himself, the administrator should intervene and kill the process for the user.

When necessary, use the UNIX **ps** command to find out information about PC-Interface UNIX processes, just as you would to find out about any other UNIX processes. Note that processes invoked via UEXEC are not associated with a terminal, so you must invoke **ps** with the appropriate option to display status information for *all* processes **(ps-ef)**.

For example, assume that a user with the user name **fred** has invoked a process **/u/fred/bin/doit** that needs to be killed. To find out the process number that needs to be killed, invoke **ps** with the appropriate option. Among the displayed data, you might see lines similar to this:

```
UID  PID  PPID  C  STIME  TTY  TIME  COMMAND
fred 346  345   1  Apr 7  ?    9:48  /u/fred/bin/doit
```

From this display you can see that the process you need to kill is number 346.

System Backup Procedures

The presence on the UNIX host of the PC-Interface host software and files created by PC-Interface users does not affect standard system backup procedures. Individual PC-Interface users can, of course, use their personal computers to backup files from the UNIX host to their own local disks. However, such procedures should not be used as a substitute for regular backups of the entire UNIX file system done by the system administrator.

Copy Protection

You may have occasion to assist PC-Interface users who have violated the restriction against simultaneous use of multiple PC-Interface disks with the same serial number. The following paragraphs describe the PC-Interface copy protection system and recommend corrective procedures to be used when a user's personal computer is disabled by the copy protection mechanism.

Using PC-Interface simultaneously from two personal computers requires two PC-Interface disks created from two different licensed PC-Interface distribution diskettes. Users may make backup copies of their distribution diskettes or working diskettes, but they may not use more than one copy at a time. **It is the system administrator's responsibility to ensure that PC-Interface is used correctly.**

Copy protection prevents the illegal duplication and use of PC-Interface software. It is illegal to simultaneously operate two or more personal computers with the PC-Interface software derived from one licensed distribution diskette. Each diskette is distributed with some unique data in the BRIDGE.SYS file that must be present in the file for correct operation.

When the PC-Interface system detects two personal computers simultaneously using PC-Interface disks created from the same distribution diskette, the second personal computer to log in to PC-Interface is halted, with a displayed message stating that a copy protection violation has occurred. This condition can be recovered from by powering off the personal computer and rebooting.

PC-Interface Error Messages

The PC-Interface host software responds to all serious and fatal PC-Interface error conditions by displaying an error message on the host system console and mailing the message to root. Refer to Appendix A of this guide for a list of these messages and recommended recovery procedures.

Tailoring Your PC-Interface System

As soon as you install your PC-Interface system, users can begin to enjoy the UNIX host resources made available by PC-Interface file service, terminal emulation, and the special PC-Interface utilities. No special configuration of the PC-Interface system is required. As you become familiar with PC-Interface operation, however, you may want to tailor the system to accommodate the way you and the system users prefer to use the system. This section offers tips on the following topics:

- **Managing Host System Disk Resources and Load:** issues concerning the host resources required for PC-Interface use.
- **Establishing Shared Personal Computer Software Libraries:** how to store DOS utilities and application programs on the UNIX host.
- **UNIX Permission Modes:** how to alter the default read, write, and execute permissions assigned to files created with PC-Interface.
- **Enabling and Disabling Terminal Emulation Ports:** how to increase or decrease the number of terminal emulation sessions supported by your host.
- **DOS Software Compatibility Issues:** technical requirements for DOS software to operate concurrently with PC-Interface.

Managing Host System Disk Resources and Load

Although the PC-Interface system requires no special procedures for managing host system disk space, you should advise inexperienced users that they are sharing a disk with other people and that they should conserve disk space. If your PC-Interface system uses multiple hosts, you can use them to your advantage by distributing user accounts among the different hosts so that resource consumption (system load and disk space use) on the different machines is about equal.

The task of load management on PC-Interface hosts is essentially the same as for any UNIX machine. Every PC-Interface user puts some load on the host, just as each user who is logged directly into UNIX does. Terminal emulation puts more of a load on the host than file service, so you may want to limit the number of simultaneous terminal emulation sessions that are allowed. (See "Enabling and Disabling Terminal Emulation Ports," later in this chapter.)

One characteristic of PC-Interface worth remembering is that it allows users the flexibility of alternating between local work (done on a local diskette or fixed disk) and remote work (done on the UNIX virtual drive). The load on the remote UNIX machine is heavier, of course, when PC-Interface users are using the virtual drive. Depending on the requirements of your installation, you may either encourage or discourage the use of the virtual drive for DOS programs, which can be run either from the user's local disk or from the virtual drive (assuming the programs are installed on the virtual drive). For example, to discourage use of the virtual drive for DOS programs, you can limit the number of publicly accessible directories containing DOS programs. (Individual users can still copy DOS programs to their own directories on the UNIX disk, however.)

Establishing Shared Personal Computer Software Libraries

The PC-Interface system administrator is responsible for creating and administering libraries of DOS software used by the network-wide community of PC-Interface users. The software libraries managed by the system administrator are analogous to personal libraries created on the UNIX disk by individual PC-Interface users to store their own copies of personal computer software. The difference is that the shared libraries created by the system administrator are accessible to all system users rather than to only a single user.

To create a publicly accessible library of DOS software on the UNIX disk, follow these procedures:

1. Choose the DOS software you wish to store and execute on the UNIX disk. PC-Interface utilities such as DOS2UNIX, EM, PRINTER, UCHMOD, UDIR, UREN, and UNIX2DOS, for example, can be installed on the UNIX disk. (These utilities are distributed with the PC-Interface personal computer software.) You can also install on the UNIX disk any other internally developed, site-licensed, or public domain DOS software that can be stored on and executed from a fixed disk. Such DOS programs might be obtained off-the-shelf or created with any of the widely available utilities that run under DOS for writing, compiling, and linking DOS programs.[1]

2. Log into your host UNIX system from a personal computer using PC-Interface.

3. Make a directory on the UNIX disk to store the software you have selected. This directory must be easy to find and execute for all PC-Interface users. We recommend **/usr/pci/pcbin** as the standard directory for DOS executable programs.

[1] Some copy-protected DOS programs cannot be installed on the UNIX disk. Such programs are generally not internally developed, site-licensed, or public domain, however, and are therefore illegal to install in a publicly accessible UNIX directory.

4. Install the DOS programs in **/usr/pci/pcbin** on the UNIX virtual drive just as you would install them in a subdirectory on a local fixed disk. During the installation procedure, refer to the virtal drive by the name PC-Interface returns to you when you log in (usually C: or D:).

5. Set the UNIX permission modes for the DOS files and the directory containing them so that PC-Interface users can access the files. The DOS files must be readable, but need not have UNIX execute permission. Directories must be readable and executable. Write permission is also often required for directories containing DOS applications, since many applications create temporary files when executed. You can use the PC-Interface UCHMOD command to set the modes correctly. To make a file readable by all other users but writable only by you, type:

   uchmod 644 **filename**

   To make the directory readable, writable, and executable by you and all other users, type:

   uchmod 777 **directory**

6. Inform PC-Interface users of the location of the newly installed software and be prepared to assist them in setting their DOS search paths appropriately. For example, assume a PC-Interface user has a personal computer with one fixed disk and normally has a DOS search path set to C:/BIN. The PC-Interface virtual drive for this user is normally drive D:. To make **/usr/pci/pcbin** easily accessible, the user's search path should be set with the command:

   path=c: \ bin;d: \ usr \ pci \ pcbin

   The PATH command can be added to the user's AUTOEXEC.BAT file to make it effective every time the user boots the personal computer. For further information on using the AUTOEXEC.BAT file with PC-Interface, refer to the *PC-Interface User's Guide.*

Tailoring Your Software Libraries

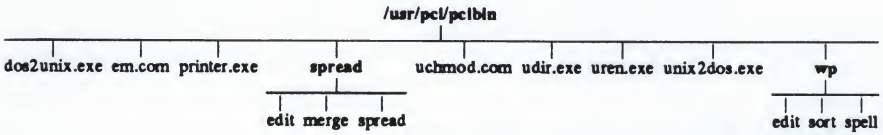
Sometimes it can be useful to create subdirectories within **/usr/pci/pcbin** so that related DOS files can be grouped together. Here are some factors to consider when evaluating the need for subdirectories of DOS software:

- A single directory containing all personal computer software is convenient because it is easily accessible to all PC-Interface users. The directory can be included in the search paths of all PC-Interface users, and there is little danger that users needing personal computer software will not be able to find it.

- A single directory containing all personal computer programs can become unmanageable if it contains a large number of personal computer programs. For example, a single personal computer application program can contain several files (sometimes including both executable and data files). If many such programs are in the same directory, it can be difficult to keep track of which files belong to which programs. Moreover, with a large number of files in the same directory, there is a risk of file name conflicts between existing files and new ones that you want to add.

- To reduce confusion, you can create separate directories for individual personal computer programs or groups of related programs. When you add such directories, however, bear in mind that users must be informed of the existence of these directories and add them to their search paths.

As an example, consider the following use of the **/usr/pci/pcbin** directory:



In this example, separate subdirectories are created for a word-processing program **(wp)** and a spreadsheet program **(spread)**. The subdirectories allow the files for the two different applications to be grouped separately. Since both applications have files named **edit**, using two subdirectories also avoids a file name conflict. The standard PC-Interface utilities remain immediately subordinate to **/usr/pci/pcbin.** For this example, DOS search paths for PC-Interface users would be set to include **/usr/pci/pcbin, /usr/pci/pcbin/wp,** and **/usr/pci/pcbin/spread.**

Software Libraries for PC-Interface UNIX Commands

The directory **/usr/bin** is recommended for storing utilities executable under UNIX that are of interest to PC-Interface users. The UNIX executable versions of **dos2unix** and **unix2dos** are installed in this directory during the PC-Interface host software installation. You can add any other UNIX programs you choose to this directory. In general, the same tailoring issues apply as for libraries of DOS software. We do not recommend creating subdirectories within **/usr/bin,** however. Following are considerations for UNIX software libraries that do not apply to DOS libraries:

- UNIX programs must be executable, unlike DOS programs which need only be readable.
- Search paths for UNIX programs are relevant only for users who log in as UNIX users—that is, users logging in from terminals and PC-Interface terminal emulation users. Do not set UNIX search paths for PC-Interface file service users.

UNIX Permission Modes

PC-Interface supports the standard UNIX file permission modes. Both file service and terminal emulation users have the same privileges and restrictions regarding access to host files as other UNIX system users.

The users **umask** determines the permission mode with which a UNIX file is created. Users connected directly to the UNIX system use the default UNIX **umask,** or they can change their own **umask** to reflect their preferred default permission modes. The PC-Interface **umask** for all RS-232 users (both file service and terminal emulation) and for Ethernet terminal emulation users is determined in the same way: they either use the default UNIX **umask** or specify their own **umask.**

PC-Interface file service users connected to the host via Ethernet all have the same **umask,** which cannot be changed by individual users. The system administrator, however, can change the system-wide **umask** that affects these users.

The script **/usr/pci/bin/pcistart** sets the **umask** when PC-Interface is initialized. This **umask** is inherited by the connection server (**/usr/pci/pciconsvr**) and applies to all PC-Interface Ethernet file service users. The initial **umask** is 2, which means:

- Files are readable by all system users and writable by both the owner and others in the owner's group.
- Directories are readable and executable by all system users and writable by both the owner and others in the owner's group.

The following example displays the contents of a typical UNIX directory with files and subdirectories created using PC-Interface file service over Ethernet. The list is produced with the **ls -l** command. Read, write, and execute permissions are shown in the first column.

```
-rw-rw-r--   1 jones    usr      3400 Mar 13 11:09 bdgtmemo
drwxrwxr-x   3 jones    usr       436 Mar  7 10:21 chapter.one
-rw-rw-r--   1 jones    usr      2400 Mar 18 16:31 message
drwxrwxr-x   2 jones    usr        92 Mar 18 16:32 pci
-rw-rw-r--   1 jones    usr      1465 Mar  4 10:56 prjctmem
drwxrwxr-x   3 jones    usr       228 Mar 18 18:43 report
-rw-rw-r--   1 jones    usr      3278 Mar  7 10:53 staff
-rw-rw-r--   1 jones    usr      2400 Mar  5 11:06 status.rep
```

To change the **umask** for PC-Interface users:

1. Stop the execution of PC-Interface.
2. Modify the file **/usr/pci/bin/pcistart** to change the line that reads:

   umask 2

   to reflect your desired **umask.**
3. Restart PC-Interface.

There are no special PC-Interface considerations associated with UNIX groups.

For further information on permission modes and groups, refer to the descriptions of **chmod, chown, umask, group,** and **setuid** in your host system documentation and the discussion of UCHMOD in the *PC-Interface User's Guide.*

Enabling and Disabling Terminal Emulation Ports

When you install the PC-Interface host software, four terminal emulation ports are enabled by default. You can reconfigure your host to support more or fewer terminal emulation sessions to accommodate changing system requirements.

To disable existing terminal emulation ports, you simply comment out the line in the **/etc/inittab** file that defines the terminal emulation port. To enable additional terminal emulation ports you either remove the comment character (if the port has already been defined) or make a new entry in the **/etc/inittab** file.

Disabling Terminal Emulation Ports

Follow these procedures to reduce the number of available Ethernet terminal emulation ports.

1. Log into the UNIX host as root or become superuser.

2. Invoke any editor to edit the file **/etc/inittab.**

3. Locate the string "PCI emulation ports." This string is followed by a line for each terminal emulation port. Change **respawn** to **off** within each line for each port being disabled.

   For example, change:

   p0:2:respawn:/etc/getty ttyp0 VT100

   to

   p0:2:off:/etc/getty ttyp0 VT100

4. Save the updated file.

5. Make the changes effective by typing:

   init q

Enabling Terminal Emulation Port

To add more Ethernet terminal emulation ports follow these procedures:

1. Log into the UNIX host as root or become superuser.

2. Invoke any editor to edit the file **/etc/inittab.**

3. Locate the string "PCI emulation ports".

4. If a port has previously been disabled by changing **respawn** to **off** within a line in the **/etc/inittab** file, change **off** to **respawn** within the line for each port being enabled.

   For example, change:

   p0:2:off:/etc/getty ttyp0 VT100

   to

   p0:2:respawn:/etc/getty ttyp0 VT100

5. If the port has not previously been defined, enter a new line in the **/etc/inittab** file for each additional terminal emulation port. Each line must have the following format:

   pn:2:respawn:/etc/getty ttypn VT100

   where n is the unique terminal emulation port number.

6. Save the updated file.

7. Make the changes effective by typing:

   init q

DOS Software Compatibility Issues

Most DOS software is compatible with PC-Interface operation. This means that DOS applications running on personal computers can access files on the UNIX host through PC-Interface. DOS applications can also be stored on and executed from the UNIX disk, as described under "Establishing Shared Personal Computer Software Libraries."

Following is a summary of the constraints on DOS software when used with PC-Interface. In general, applications will operate with PC-Interface if they do not:

1. Overlay the DOS or BIOS area of storage.

2. Program the 8259 interrupt controller in a way that interferes with PC-Interface's use.

3. Disable interrupts, fail to issue an end-of-interrupt of IRET on a hardware interrupt level, or mask selected interrupt levels for more than 100 ms.

4. Use interrupts 13, 25, or 26 for access to the virtual drive.

5. Configure hardware device registers that belong to the network hardware used by PC-Interface.

6. Make incorrect use of timer interrupts.

7. Open more than 127 files (that is, require the presence in the CONFIG.SYS file of a FILE=n command, where n is greater than 127).

This list of limitations for operation may not be complete.

There are a few commands and functions in DOS Version 3.0 and higher that are not fully supported by PC-Interface. These commands and functions are:

1. JOIN command

2. SUBSTITUTE command

3. Function 3DH (open a file) functionality added with DOS Version 3.0

4. Function 5CH (lock/unlock file access)

In addition, we do not recommend using the following:

- Any version of IBM PC-DOS print spooling.

- The MS-DOS or PC-DOS ASSIGN command with the virtual drive.

- Version 2.11 MS-DOS or PC-DOS COMP command.

Chapter 5. PC-Interface System Architecture

Introduction

This chapter provides an architectural overview of the PC-Interface system. It defines the purpose of each software module on both the personal computer and the UNIX host and describes how the modules interact during PC-Interface file service and terminal emulation sessions. First, the software modules on both the personal computer and the host are listed along with a brief statement of their purpose. Following this list is a discussion of PC-Interface operation over Ethernet and RS-232.

PC-Interface Software Modules

Personal Computer Modules

The following PC-Interface modules operate on the personal computer under DOS:

PCIINIT.EXE: Initializes PC-Interface.

BRIDGE.SYS: Provides basic PC-Interface support.

CONFIG.SYS: Identifies device drivers to be loaded by DOS.

LED.SYS: Ethernet device driver.

LOGIN.EXE: Establishes a host connection for host file services.

LOGOUT.EXE: Terminates a host file services session.

DOS2UNIX.EXE: Converts text files in DOS format to UNIX format.

EM.COM: Establishes a terminal emulation session.

PRINTER.EXE: Directs output to local or remote printer.

UCHMOD.COM: Changes file permissions for files on virtual drive.

UDIR.EXE: Lists virtual drive directory contents in UNIX style.

UEXEC.EXE: Remotely executes host UNIX processed from DOS environment.

UNIX2DOS.EXE: Converts text files in UNIX format to DOS format.

UREN.EXE: renames UNIX files and directories.

Host Modules

The following PC-Interface modules operate on the host under UNIX:

/usr/pci/bin/pcistart: Starts PC-Interface daemons on the host.

/usr/pci/bin/pcistop: Terminates execution of PC-Interface Ethernet processes on the host.

/usr/pci/pcimapsvr.eth: Daemon that listens for broadcast requests for site tables from LOGIN. Also listens for broadcasts from **pciconsvrs** on the net announcing that they are alive.

/usr/pci/pciconsvr.eth: Daemon that listens for a connection request from LOGIN. Also broadcasts "pciconsvr host name here" messages, indicating that the host is available for PC-Interface service.

/usr/pci/pcidossvr.eth: Maintains an exclusive dialogue with the personal computer module BRIDGE.SYS. Executes UNIX commands on hehalf of the personal computer user and transmits the results back to the personal computer.

/usr/pci/pcidosout.eth: A process created for use during terminal emulation over Ethernet. It takes data passed from the UNIX shell via the pseudo-tty (pty) and sends it over the network to the personal computer.

/usr/pci/pcidossvr.232: Same as **pcedossvr.eth,** but works over RS-232.

/usr/pci/pcidosout.232: Same as **pcidosout.eth,** except it is used for terminal emulation over RS-232.

/usr/pci/pciptys: A text file listing pseudo-ttys (ptys). Used for terminal emulation.

/usr/pci/errlogger: A utility invoked automatically when PC-Interface encounters a serious or fatal error. It sends mail to root reporting the problem and also displays the message on the host console.

/usr/bin/unix2dos: Converts text files in UNIX format to DOS format.

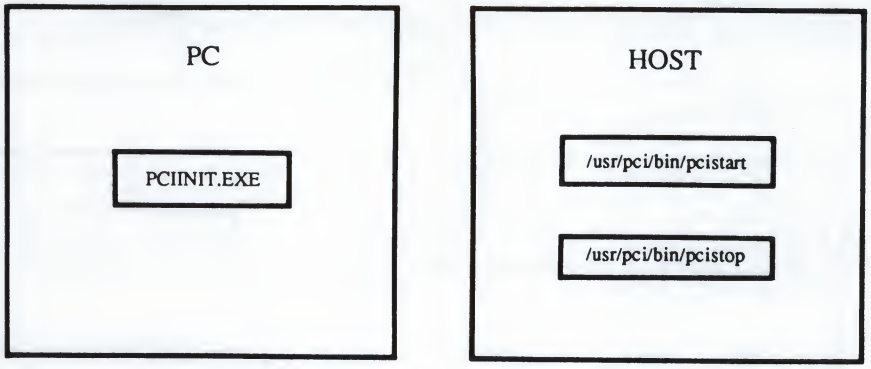
/usr/bin/dos2unix: converts text files in DOS format to UNIX format.

/usr/pci/bin/pcidebug: A utility for logging detailed information about PC-Interface transactions in the directory **/usr/spool/pcilog.** This utility is for debugging and should be used only under the direction of PC-Interface customer-support personnel.

/usr/spool/pcilog: A directory that stores the debugging information generated by **/usr/pci/bin/pcidebug.**

Software Structure for PC-Interface Over Ethernet

Starting and Stopping PC-Interface Software on Personal Computers and Hosts



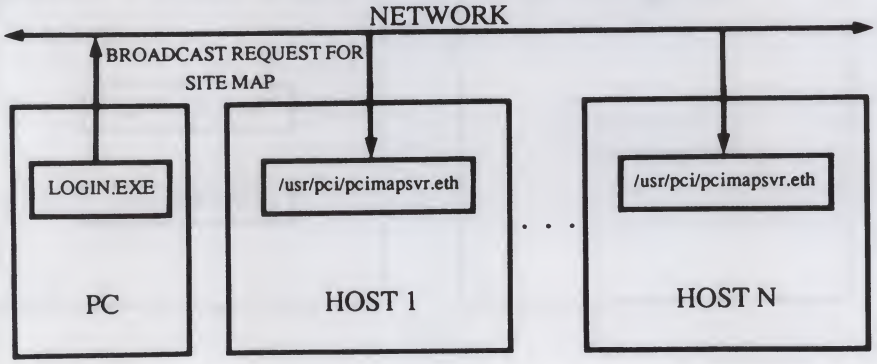
Before any interaction between personal computers and hosts can take place using PC-Interface, the PC-Interface software must be installed and initialized on both the personal computer and the host. The PCIINIT.EXE program on the personal computer initializes the personal computer side of PC-Interface. PCIINIT is normally invoked from the user's AUTOEXEC.BAT file, so it is executed automatically when the personal computer is booted.

Modules that support PC-Interface service on the personal computer are loaded as device drivers at boot time. These modules are specified in CONFIG.SYS. When Ethernet is used, these modules are BRIDGE.SYS and LED.SYS. BRIDGE.SYS is the basic PC-Interface support module. LED.SYS is the Ethernet driver. All communication originating from the personal computer side of PC-Interface and destined for the network passes through the Ethernet driver.

The module **/usr/pci/bin/pcistart** initializes PC-Interface on the host. Its purpose is to start the two PC-Interface daemons that run on the host—**/usr/pci/pcimapsvr.eth** and **/usr/pci/pciconsvr.eth.** The **pcistart** module also establishes the UNIX environment in which PC-Interface runs.

This module **/usr/pci/bin/pcistop** stops the two daemons and terminates the execution of PC-Interface Ethernet processes on the host. (Note that no equivalent module is required on the personal computer, since operation on the personal computer side is normally terminated by turning off the personal computer or booting without PC-Interface.)

Establishing a PC-Interface Connection Between a Personal Computer and a Host



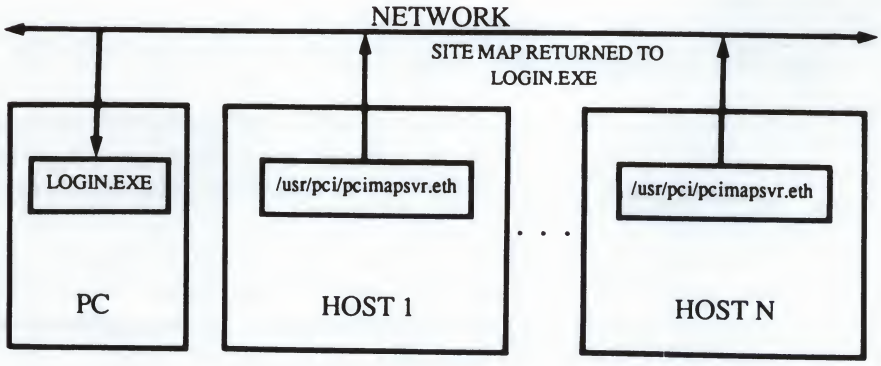
To begin a PC-Interface file service session, the personal computer user executes the LOGIN command. LOGIN.EXE (LOGIN) broadcasts over the network a request for a site map. A site map is a table listing the hosts on the network that are up and running PC-Interface—that is, all the hosts which the personal computer can possibly connect.

The daemon **/usr/pci/pcimapsvr.eth (pcimapsvr)** on each host has two related functions:

1. To listen for broadcast "pciconsvr **host name** here" messages from each host running PC-Interface. (The module that broadcasts the message is **/usr/pci/pciconsvr.eth.**) Each **pcimapsvr** keeps a table of available hosts that is updated approximately every 30 seconds based on these broadcasts.

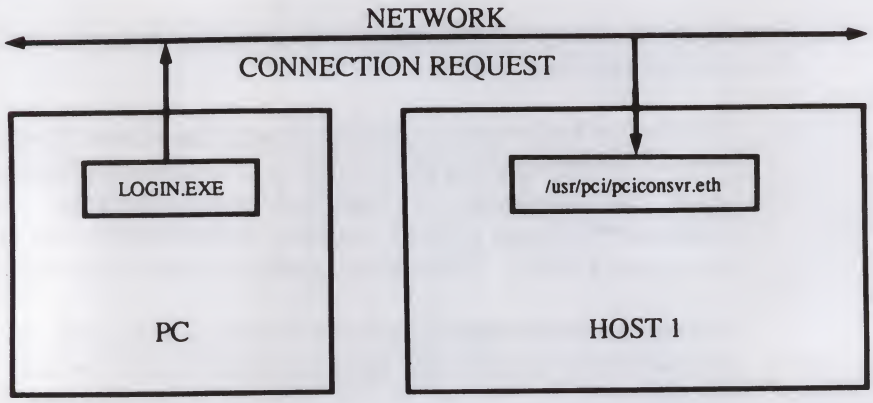
2. To listen for broadcast requests for site maps issued by LOGIN and send the current site map to any LOGIN process that requests it.

In response to LOGIN's request for a site map, therefore, each **pcimapsvr** returns its current site map, as illustrated in the following diagram:



On the personal computer side, LOGIN listens for the first returned site map and ignores any other incoming site maps from other hosts.

LOGIN displays for the personal computer user the list of available host. The user selects a host from the list, and LOGIN formats and sends a connection request to **/usr/pci/pciconsvr.eth (pciconsvr)** on the selected host, as illustrated in the following diagram:



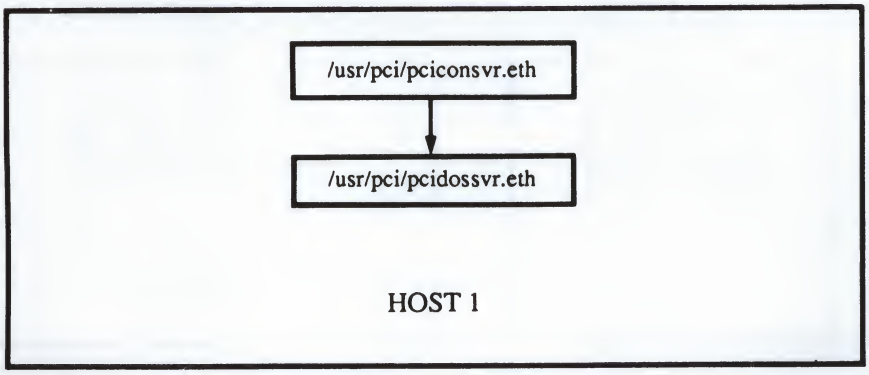
Note that the connection request need not necessarily be directed to the same host that provided the site map to LOGIN. Since all **pcimapsvrs,** under normal circumstances, have a current list of all available hosts, LOGIN can use the site map provided by any **pcimapsvr.**

Like the pcimapsvr, each **pciconsvr** has more than one job:

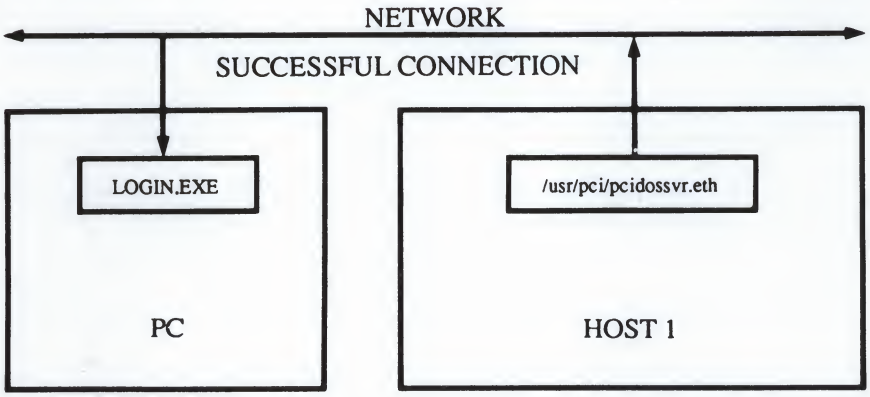
1. To broadcast "pciconsvr **host name** here" messages approximately every 30 seconds. These messages are used by **pccimapsvrs** to keep their lists of available hosts up-to-date.

2. To manage the connection between the personal computer and the host. This task includes several subordinate tasks. The connection management process begins when a **pciconsvr** receives a connection request, as illustrated above.

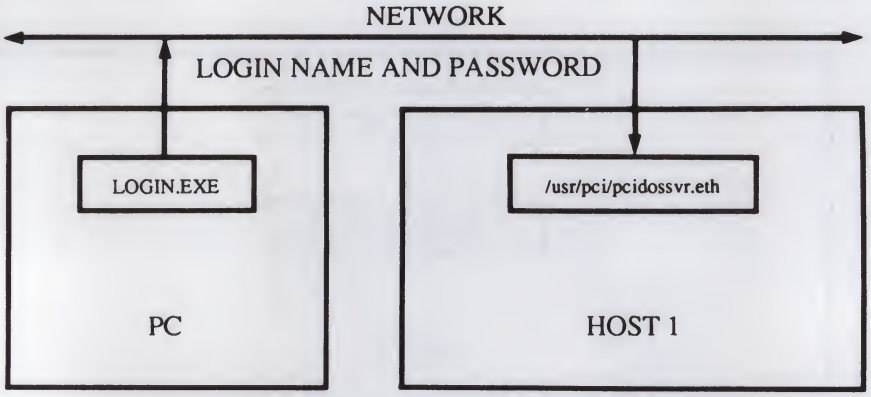
When the **pciconsvr** receives a connection request from LOGIN, it assigns a port number for use in subsequent commuication with the personal computer, and spawns a new process, **/usr/pci/pcidossvr.eth (pcidossvr):**



The **pcidossvr** inherits the port from the **pciconsvr** and informs LOGIN that a connection was successfully established, as illustrated in the diagram:



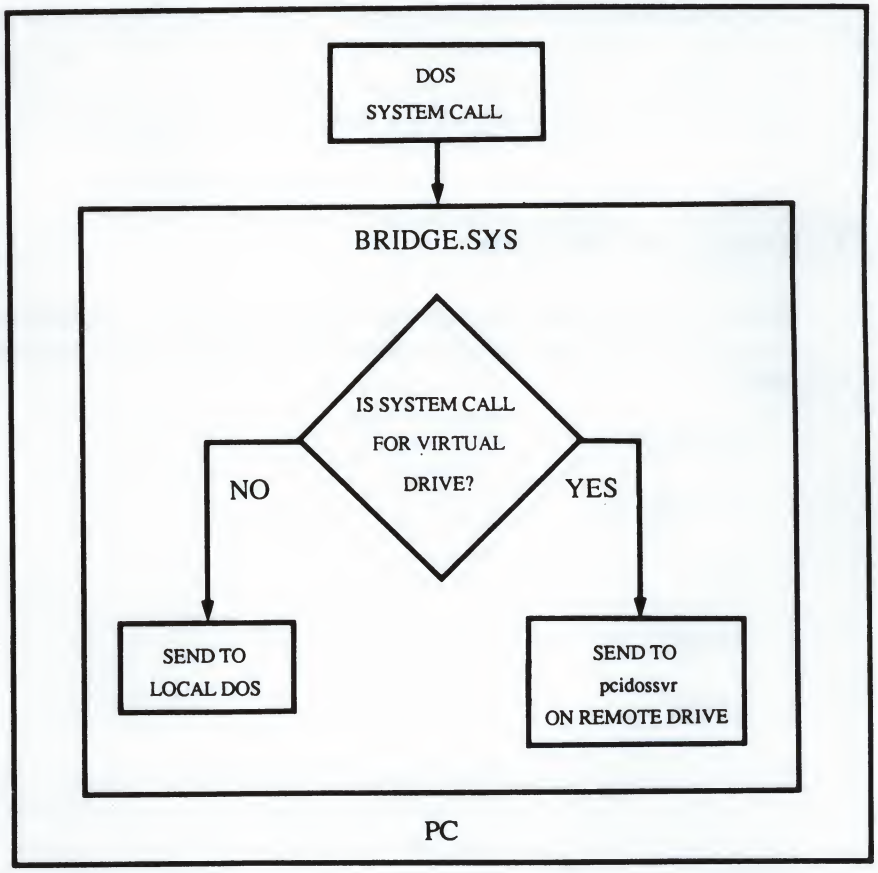
LOGIN then prompts the user for user name and password and transmits them to the **pcidossvr,** as the following diagram illustrates:



The **pcidossvr** validates the user name and passward, changes its own user and group IDs to those of the logged in user, and informs LOGIN that the process has been successsfully completed. LOGIN tells the user on what drive the UNIX file system is available, welcomes the user to PC-Interface, and returns the DOS prompt to the user.

Using an Established PC-Interface Connection

At this stage, bidirectional communication can occur between the personal computer and the host. On the personal computer side, the BRIDGE.SYS module takes over the task of communicating with the remote host. A top-level view of the operation of BRIDGE.SYS is as follows:



As shown in the illustration above, BRIDGE.SYS is interposed between the user's invocations and the local DOS that the user booted on the personal computer. The BRIDGE.SYS module intercepts all DOS system calls (INT 21H). If the system call is a request for a local device, BRIDGE.SYS passes it unmodified to the local DOS. If the request is for the remote (virtual) drive, BRIDGE.SYS formats a transaction and sends it to its **dossvr** on the UNIX host to be processed.

The **pcidossvr** module on the remote UNIX host translates the user's standard DOS system calls into appropriate UNIX commands, executes them on behalf of the user, translates the results back into standard DOS, and returns the results to the user via BRIDGE.SYS.

PC-Interface Utilities

When a PC-Interface file service session has been established as described above, the following special PC-Interface utilities become useful:

- PRINTER
- UCHMOD
- UDIR
- UEXEC
- UREN

These commands allow the user to manipulate and display data on the remote UNIX drive in ways not allowed by standard DOS. For description of their operation, see the *PC-Interface User's Guide*. The following PC-Interface commands can be used independently of file services, as well as during a file service session:

- **/usr/bin/dos2unix** (UNIX executable) and UNIX2DOS.EXE (DOS executable)

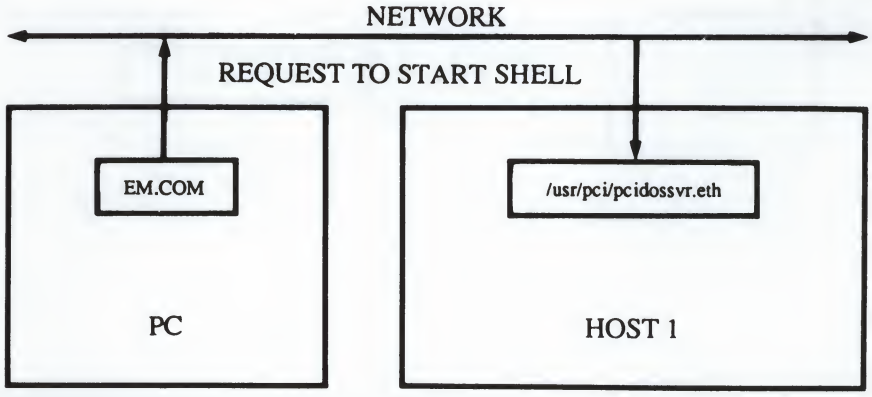
- **/usr/bin/unix2dos** (UNIX executable) and DOS2UNIX.EXE (DOS executable)

- EM.COM

For a description of both the DOS and UNIX executable versions of DOS2UNIX and UNIX2DOS, refer to the *PC-Interface User's Guide*. EM.COM is described from the user's point of view in the *PC-Interface User's Guide* as well, but additional relevant information appears below.

Using Terminal Emulation

When the PC-Interface user wants to communicate with the remote UNIX system through a standard UNIX shell rather than through DOS commands, the terminal emulation program EM.COM (EM) is used. When EM is invoked, it broadcasts a request for a site map, just as LOGIN does. The user is returned a site map and chooses the host on which the terminal emulation session is to be established. EM then requests the **pcidossvr** on the remote host to start a UNIX shell:

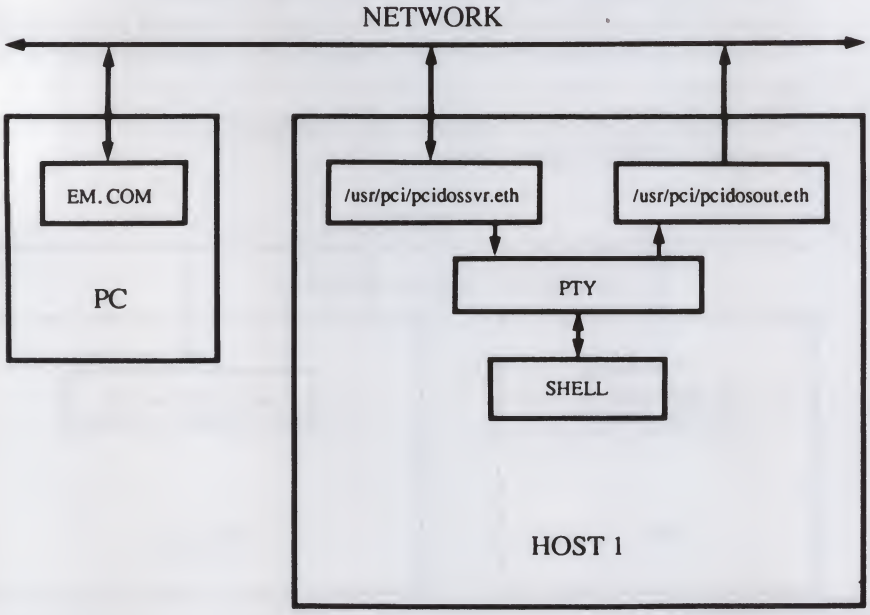


The UNIX shell is presented to the personal computer user through a pseudo-tty (pty) on the UNIX side.

The events that occur when the **pcidossvr** receives the request for EM to start a UNIX shell are as follows:

- The **pcidossvr** opens the control side of the pty. To do this, the **pcidossvr** attempts to open each pty listed in the file **/usr/pci/pciptys** until it succeeds in opening one.
- Once the control side of the pty is opened, **getty** starts a login process on the slave side of the same pty.
- The **pcidossvr** spawns another process, **/usr/pci/pcidosout.eth (pcidosout),** which is responsible for sending characters from the UNIX shell (via the pty) back to the user.

The following illustration depicts the end result of this process:



The **pcidosout** process is required because the **pcidossvr** cannot listen for PC-Interface file service messages from the network at the same time as it listen for characters sent from the pty. The **pcidosout** process is therefore assigned the task of reading characters from the pty and sending them back to the user.

When EM is invoked during a PC-Interface file services session, the terminal emulation process does not terminate file service. The user can alternate freely between standard PC-Interface file service and terminal emulation. Data sent by EM to the remote host is marked as terminal emulation data and is processed as described above. To suspend terminal emulation (without permanently terminating the emulation process), the user types F10. This allows the user to resume the file service session. To return to terminal emulation, the user reinvokes EM and is returned to the ongoing emulation session.

To terminate the emulation session, the user types F7. A request to terminate emulation goes to the **pcidossvr**, which kills the **pcidosout** process. The **pcidossvr** closes the control side of the pty. The pty driver sends a SIGHUP signal to the UNIX processes associated with that pty. The UNIX shell closes the slave side of the pty and exits. The user can then resume PC-Interface file service.

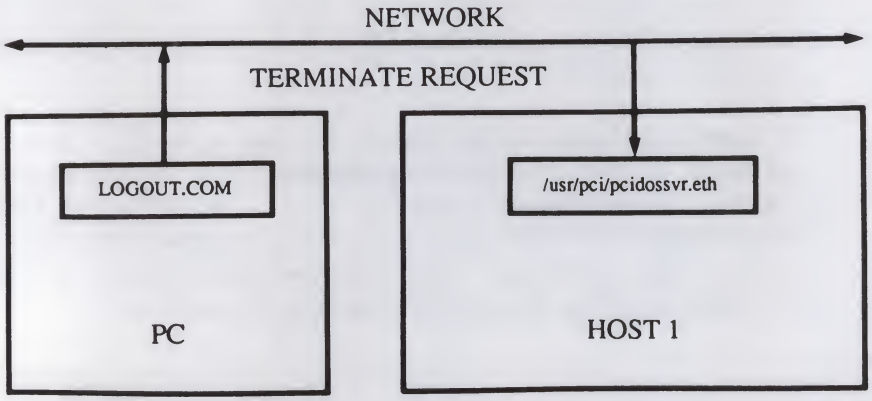
Using the Terminal Emulator Without Establishing a File Services Session

The terminal emulator can be used without first establishing a file service session—that is, without logging into PC-Interface. PC-Interface must be initialized (using PCIINIT.EXE) for emulation to work, however.

When EM is invoked independently of a file services session, it broadcasts a request for a site map, the same as it does when invoked during a file services session. When the user chooses the desired host for terminal emulation, EM invokes a **pcidossvr** for the user since, without a file service session, there is no **pcidossvr** already executing on behalf of the personal computer user on the remote host. The user can then log into the UNIX host and use terminal emulation as described above.

Ending a PC-Interface File Services Session

When the personal computer user executes the PC-Interface LOGOUT.COM command, a terminate request is sent to the **pcidossvr,** as illustrated in the following diagram:



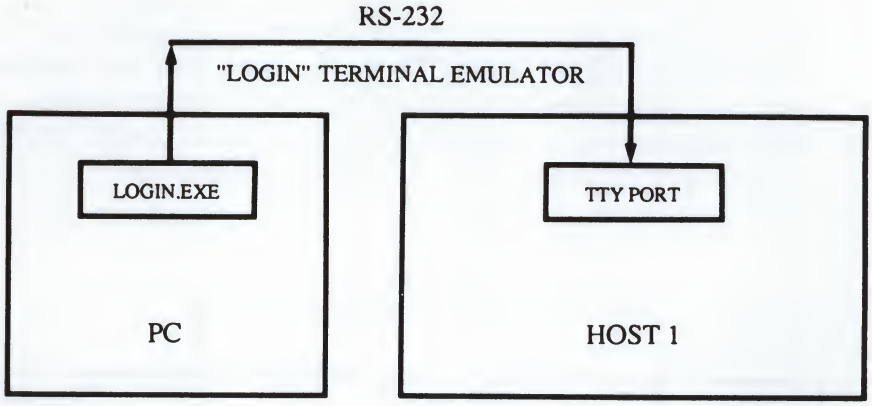
The **pcidossvr** sends a terminate request signal to any **pcidosout** process that might still exist and then exits. When the **pcidosout** and **pcidossvr** processes exit, the **pciconsvr** frees the network port and cleans up its status tables.

Software Structure for PC-Interface Over RS-232

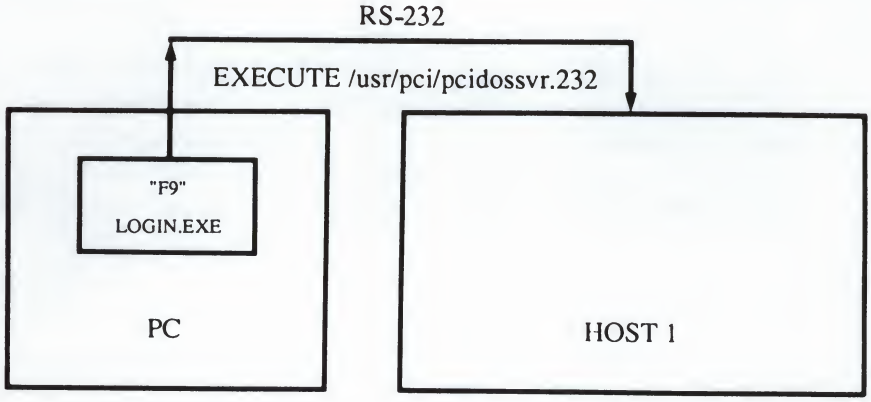
To use PC-Interface over RS-232, the PC-Interface software on both the personal computer side and the host side must be installed and initialized just as in the Ethernet case. (PCIINIT.EXE is used on the personal computer and **/usr/pci/bin/pcistart** is used on the host.) BRIDGE.SYS is loaded on the personal computer side, as in the Ethernet case. No device driver (such as LED.SYS) is required, however.

When the user invokes the LOGIN command, a site map is requested and returned to the user, again as in the Ethernet case. The list of available hosts displayed for the user by LOGIN identifies the RS-232 line as one of the available "hosts." (If Ethernet is not also available to the user, no other hosts are listed.)

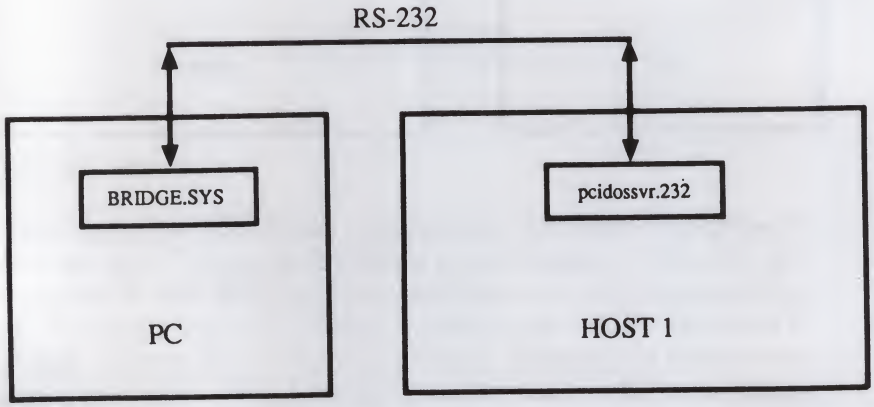
No **pciconsvr** is used with RS-232. Instead, when the user selects the RS-232 line, LOGIN creates a special-purpose terminal emulator specifically for the purpose of logging into the remote host, as illustrated below:



The RS-232 cable is connected to the UNIX host just like any RS-232 cable coming from a standard terminal. Using the special purpose emulator, the user logs in to the UNIX host as though from a terminal. When the login is finished, the user types F9, which terminates the special-purpose emulator and invokes **/usr/pci/pcidossvr.232** on the host, as illustrated in the following diagram:



As in the Ethernet case, LOGIN then tells the user on which drive the UNIX file system is available, welcomes the user to PC-Interface, and returns the DOS prompt the user. (Unlike the Ethernet case, the **pcidossvr** for RS-232 does not have to validate the user name and password or change its user and group IDs. Because the user has logged directly into the UNIX system as a standard UNIX user, the UNIX login process validates the user, and the **pcidossvr** automatically inherits his user ID and group ID.) At this stage, the user can communicate with the remote host and use PC-Interface utilities just as in the Ethernet case, as shown here:



The software structure and operation for terminal emulation over RS-232 is the same as for terminal emulation over Ethernet with one exception: terminal emulation is not available concurrently with file services. The user cannot be logged into a PC-Interface file services session over RS-232 when the emulator is invoked over RS-232.

Chapter 6. Some Problem-Solving Tools for PC-Interface

Introduction

As system administrator, you may be called upon to assist users with problems they encounter using PC-Interface. The problem determination procedures in this chapter can help you determine if a problem is hardware related, software related, or caused by user error.

For hardware problems, refer to the *Technical Reference* or *Guide to Operations* for the user's personal computer. For user error and software problems that are not described in this chapter, refer to the *DOS Reference Manual* or *User's Guide*. For problems with the network connection, refer to Appendix D, Troubleshooting; or Chapter 4, Data Link Services—Specifically the dlstat and dlmon utilities; in the *OfficeLan Administrator's Guide*. This chapter concentrates on errors that are attributable to PC-Interface.

The approach to problem determination taken by this chapter is to break down the problem into its components and address the individual problems in order.

During this process you need to carefully collect the clues provided by the system (such as error messages and directory contents) to determine the exact nature of the problem.

Before using the procedures in this chapter, try to reproduce the problem you are having on a different personal computer. Make sure that you use the same software and hardware environment in running your duplication test.

If you cannot reproduce the problem, it may have been caused by a loose connection between the personal computer and the network or by hardware problems internal to the personal computer. If ensuring that the network is connected properly does not correct the problem, your personal computer may need repair. If the problem presists on another personal computer, continue with this procedure to classify your problem.

Most PC-Interface problems occur in one of four major areas: system startup, establishing communication, proper usage, and normal operation. You can determine in which area to begin problem determination by following this procedure:

1. Does the failure occur during loading or PC-Interface initialization, before you attempt to log in?

   Yes: Start with "Problems With System Startup."

   No: Continue with the next step.

2. Does the failure prevent a session from being established?

   Yes: Start with Problems Establishing Communication."

   No: Continue with the next step.

3. Does the failure occur during a PC-Interface session?

   Yes: Start with "Problems With Usage." If your problem is not described there, continue with "Problems With Normal Operation."

   No: The problem is not PC-Interface related. Refer to your *Guide to Operations* or specific application manual.

Refer to the proper heading in this section, as determined above, to find the cause of and solution for your problem. If you try the corrective actions described and the same problem persists report the problem to your PC-Interface vendor.

Problems With System Startup

This section helps you determine the cause of a problem that occurs during system initialization, before you attempt to log in.

The PC-Interface system must be installed on a working diskette (that is, a diskette that contains the DOS system files) or fixed disk. The installation procedure given in the *PC-Interface User's Guide* combines the required PC-Interface files on a diskette (or fixed disk) with the DOS operating system, creating a working diskette.

Attempt to load the personal computer from the working disk containing PC-Interface by turning on the personal computer or by holding the CTRL and ALT keys while you press the DEL key. When first powered on or booted, the personal computer runs a Power-On Self Test (POST). On some personal computers, a successful POST prints no messages; on others, it prints a series of messages of the form "xx KB," indicating the amount of memory that has been tested. Compare the POST when booting with your PC-Interface working diskette to a POST when booting from your DOS system disk without PC-Interface.

Use the following procedure to determine if your problem can be attributed to a system startup error.

1. Did the POST complete correctly without producing any error messages when the personal computer was powered on or booted?

   Yes: Continue with the next step.

   No: Refer to the *Guide to Operations.*

2. Was a "Non-System disk or disk error" message displayed?

   Yes: The disk does not have the correct operating system files on it for DOS to load. Refer to your *Guide to Operations* and the *PC-Interface User's Guide* for the procedures for creating a PC-Interface working disk.

   No: Continue with the next step.

3. Were the banners "PC-Interface @(#) Release x.x.x" displayed?

   Yes: Continue with the next step.

   No: The file CONFIG.SYS must be present on the root of the PC-Interface disk and must contain the following lines:

   ```
   device = led.sys
   device = bridge.sys
   ```

   (The first line can be omitted if PC-Interface is not being used with Ethernet.) There may be directories specified for the devices (that is, you may see " \ 274 \ bridge.sys").

   Repeat the PC-Interface installation process if necessary to correct this problem.

4. Was a "Bad or missing BRIDGE.SYS" or "Bad or missing LED.SYS" message displayed?

   Yes: The file BRIDGE.SYS or LED.SYS was not found where specified in the CONFIG.SYS file. Ensure that the files are present in the locations specified in CONFIG.SYS. Repeat the PC-Interface installation process if necessary to correct this problem.

   No: Continue with the next step.

5. Does the file AUTOEXEC.BAT exist in the root of the PC-Interface disk?

   Yes: The file should contain the following line:

   This line tells DOS to run PCINIT.EXE.

   If this line is missing from the AUTOEXEC.BAT file, add it using any text editor or by repeating the installation procedure.

   No: Repeat the installation process to create an appropriate AUTOEXEC.BAT file.

6. Was the message "Bad command or file name" displayed?

   Yes: The program PCIINIT.EXE was not found. It must be in a place where it can be executed at startup from the file AUTOEXEC.BAT. It must either be in the root of the PC-Interface disk, or in a subdirectory specified in the search path. Repeat the installation process to place the file in the correct location.

   No: Continue with the next step.

7. Was the message "pciinit: cannot find correct BRIDGE device" displayed?

   Yes: If the answer to step 4 was yes, this should not happen. This message is displayed, however, in the event that there was a failure in steps 3 or 4. Repeat the PC-Interface installation process to correct this problem.

   No: Continue with the next step.

8. Was the message "PC-Interface Release x.x.x Initialized" displayed?

   Yes: The resident part of PC-Interface initialization has succeeded. You should continue with the section on establishing communication.

   No: If you have verified the contents of AUTOEXEC.BAT, then PCIINIT should have been executed. If an error message was displayed (other than the ones described above), report the problem to your PC-Interface vendor. If your personal computer appears to have hung (i.e., the screen may be blank and the personal computer isn't doing anything for a few minutes), your network connection may be disconnected, loose, or faulty.

Problems Establishing Communication

This section helps you determine the cause of a problem that occurs during attempts to establish communication with your UNIX host. Depending on the type of service that you want to use, one of two programs is run to establish communication. If you want to initiate host file services, you run the LOGIN program. If you want to use your personal computer to emulate a VT100 terminal, you run the EM program.

Refer to Appendix A for errors that are not covered here.

1. The command LOGIN or EM is executed to initiate a connection. After executing LOGIN or EM, was the message "Bad command or file name" displayed?

   Yes: the LOGIN.EXE or EM.COM file is not in the current directory nor in a directory that is listed in your DOS search path. The path is normally set during system startup by AUTOEXEC.BAT. To set the path yourself, see the DOS User's Guide. If LOGIN.EXE or EM.COM is not on your disk, repeat the PC-Interface installation process.

   No: the LOGIN program should have displayed a full-screen host selection menu. The EM program should have displayed an initialization screen. Continue with the next step.

2. Was the message "PC-Interface Not Installed" displayed?

   Yes: You have a problem with system startup. Go to that section to determine the exact nature of the problem.

   No: If you are using LOGIN continue with step 3. If you are using EM, continue with the next step.

3. Does the personal computer hang after you execute EM?

   Yes: Check the current directory on the drive from which you invoked EM for an obsolete EM.SES file left from a previous, aborted session. If such a file exists, delete it and try executing EM again.

   There may be a host connection problem. Check your personal computer's connectors to ensure that you are connected to the network. Verify that the network is operational, and restart your system to try again.

   You may not have enough memory left to execute the emulator. Remove from your AUTOEXEC.BAT the invocation of memory-resident programs. Restart your system and try EM again with no other memory-resident programs installed.

   If none of the above suggestions work, report the problem to your PC-Interface vendor.

   No: Consult the error message section of this manual for any further EM errors or report the problem to your PC-Interface vendor.

4. When running the LOGIN program, was the message "Selected Host Not Available" or "Selected Host Not Available—Try Again)" displayed?

   Yes: You selected a connection path that became unavailable after it was included in the host selection menu, or your selected path cannot accommodate any more users, or there was a network communciation problem. Verify that the network is working and that your personal computer is properly connected. Try to select the same connection path again. If it fails again, try a different connection path if one is available. If that works, check that the desired host is up and running PC-Interface and that the hardware between the user's personal computer and the host is operational.

No: Continue with the next step.

5. Was the message "Login incorrect. Try Again? (y or n):" displayed?

   Yes: The user name or password that you entered at the prompts were incorrect. Check that you are using the proper user name and have the correct password. Enter y to try again.

   No: consult the error message section of this manual for any further LOGIN errors.

Problems With Usage

This section describes some specific PC-Interface interactions that may result in problems if PC-Interface is not used correctly. Use the following procedures to determine if your problem is attributable to improper usage of PC-Interface.

1. Was a message indicating that you are out of room on your disk displayed?

   Yes: If you receive this message when you leave terminal emulation without closing the session (F7), you do not have enough room in the current directory on the drive from which you invoked EM for the EM.SES file.

   Regardless of the program you were running, you must remove some files from your working disk in order to make room to save files.

   No: Continue with the next step.

2. Did you try unsuccessfully to print a file on the virtual drive? (Make sure that the printer is operational and not jammed.)

   a. If you entered PRINTER REMOTE and then used one of the following:

      - The print screen key.
      - The DOS PRINT command.
      - COPYing your file to the print device (LPT1).

did your file not print at all?

Yes: Check that the UNIX print spooler is enabled.

Try the other printing methods. If your file still doesn't print, report the problem to your PC-Interface vendor.

No: Continue with the next step.

b. Are you printing from within an application program and expecting to see the printout before you leave the program?

Yes: You need to have used PRINTER REMOTE /T, which enables the timeout option. Without this option, your data does not print until you exit your application program.

No: Continue with the next step.

c. Are you using the PRINT program and your listings are being printed with the other listings embedded in them?

Yes: This printing option should not be used with remote printing. See the *PC-Interface User's Guide.*

No: If your printing problem is not described here, consult your PC-Interface vendor.

3. Do you have strange file names with braces ({ and }) embedded in them?

Yes: File names with braces in them are usually UNIX file names that cannot be translated directly to DOS file names. Read the section in the *PC-Interface User's Guide* on naming files.

No: Continue with the next step.

4. Are there files that you can see when using EM that do not appear when using host file services, even if you run the UDIR program?

Yes: When logged in to UNIX you may use directory commands that allow you to see hidden files (files that begin with a period). These files cannot be accessed using host file services. If you wish, you can rename them while using EM to names that are visible to the DOS operating system, UDIR, and UREN. See your *DOS User's Guide* for a description of legal DOS names.

No: Continue with the next step.

5. Did you just edit a file, but when you printed it or edited it again, it was changed?

   Yes: PC-Interface has no mechanism to prevent two users from concurrently writing to a file for which they both have write permission. In addition, PC-Interface cannot prevent one user from overwriting a file that has just been updated by another user. Most DOS programs are NOT designed to allow multiple users to update files reliably.

   The usual result of concurrent updating is that the changes made by the last user writing and closing the file are preserved and those made earlier are lost. Multiple users must coordinate their efforts and serialize the updates they make.

   No: Continue with the next step.

6. Do you see a file on the virtual drive, but cannot access or write to it, or did the permissions appear to change?

   Yes: There are differences in the DOS and UNIX file permissions that can be visible to the users. This is especially true when using the UCHMOD command to change UNIX permissions to control certain types of file sharing.

   If file permissions are set in a particular way by using the UCHMOD command and then the file is updated by using almost any DOS program, the UCHMOD operation must be repeated. This is because most DOS programs perform updates by renaming the existing file and then creating a new file with the updated contents of the original file. The new file is created with default permissions according to the UNIX **umask** environment variable, which is set when PC-Interface is started up. A common default for this variable sets all permissions to read and write for files, and to read/write/execute for directories.

   No: Continue with the next step.

7. Do you see a directory on the virtual drive, but cannot access or write to it, or did the permissions appear to change?

   Yes: A confusing situation can occur when PC-Interface users access directories for which they do not have read permission. If the programs or files they are accessing are opened with a DOS open system call, the open succeeds. If you try to search

the directory for a file, the search fails. Some libraries are known to perform directory searches even when this is unneeded, which results in certain programs failing to open files if the user doesn't have read permission for the directory.

No: Continue with the next step.

8. Are you having system resource or sporadic failures like "cannot create file," "cannot open file," or "file not found," or find that you cannot execute a file on the virtual drive?

   Yes: Such failures are most commonly caused by UNIX file or directory permission conflicts. Review the "UNIX File Permissions" section in Chapter 2 of the *PC-Interface User's Guide*.

   No: Continue with the next step.

9. When using the virtual drive, do you see miscellaneous error messages that appear to be incorrect?

   Yes: Some DOS programs perform operations that cannot be supported by PC-Interface. For example, there are BIOS functions that perform read and write of physical disk blocks. This operation cannot be mapped into a corresponding operation on a host volume. Programs that attempt to do this produce an error diagnostic. Fortunately there are only a small number of programs used today that do not operate correctly under PC-Interface. Use the section on "Problems With Normal Operations" to isolate this type of error.

   No: Check Appendix A to see if the error you are investigating is explained. If not, use the section on "Problems With Normal Operations" to determine whether or not your problem is related to PC-Interface.

Problems With Normal Operations

This section addresses problems that occur during an established PC-Interface session. A procedure is provided in this section for determining if your problem is indeed a PC-Interface problem.

If your problem is with one of the utilities provided with PC-Interface (other than PCIINIT, LOGIN and EM), read the appropriate section in the *PC-Interface User's Guide* that describes the utility. PC-Interface internal problems are the main focus of this section.

Most PC-Interface operation problems fall in two main groups—problems that are caused by PC-Interface and problems that are the result of the presence in memory of PC-Interface. The basic approach to isolating your problem is to re-create the conditions that produced the problem, but without the PC-Interface software being present. If this test passes, try again with PC-Interface present but not active. Use the procedure below as a guide for doing these tests.

Before you continue, carefully note the steps that led up to the appearance of the problem when running your application program. You will be repeating these steps later.

1. Copy all the files that were used when the problem occurred to a DOS disk. This may include data files, DOS programs, and configuration files. The DOS disk can be a diskette or a fixed disk. We refer to this disk as the test volume.

2. Remove PC-Interface from your working disk, as follows:

   a. Make a copy of CONFIG.SYS, so that you don't lose your original file. Using any text editor, remove the following two lines from the end of the CONFIG.SYS file:

      device = led.sys
      device = bridge.sys

      There may be directories specified before the file names (e.g., you may see "\274\bridge.sys"). Remove the two lines entirely. (Note that the first line may not be present if PC-Interface is not being used with Ethernet.)

b. Make a copy of AUTOEXEC.BAT, so that you don't lose your original file. Edit AUTOEXEC.BAT using any text editor and remove the following line:

```
pciinit
```

3. Fix the path or environment.

   If the path or any environment variables are set up to make use of the virtual drive, they must be changed to reflect the locations of the files that you moved to the DOS test volume. This may require changes to AUTOEXEC.BAT or other configuration files, or changes to the manual procedure that you go through to run the program. Below are some examples of things that are likely to change. There may be other changes that are not listed below but are specified in the application program manual.

   a. Diplay your path. Does your path contain references to the virtual drive?

      Yes: Set your path (using path=n:dir1;m:dir2. . . , where n and m are drive specifiers not on the virtual drive, and the dir entries are directories where you want the command line processor to search).

      No: Continue with the next step.

   b. Depending on your application program, you may need to have all the data files in the same directory as your application program. This directory may need to be on your current drive. Are all programs and data files for this test located on the test volume so that they can be found using the path?

      Yes: Continue with the next step.

      No: Correct the path or move the program or data files appropriately.

4. Now restart DOS without PC-Interface and confirm that it is not present by noting whether there are any banners printed during the load process that say "PC-Interface." If any such banners appear, something was not done in the steps above and should be corrected before proceeding.

5 Now perform the sequence you noted earlier that produced the error. Is the error still present?

Yes: The error has nothing to do with PC-Interface, and the user should determine if the error is in the application program or in DOS itself. This is best done by referring to the trouble-shooting sections of the application program manual or by referring to the *Guide to Operations* for your personal computer.

No: If the error does not occur, the problem is either in PC-Interface or because PC-Interface is present but not actually responsible for the error. The latter case can occur if programs or devices disregard DOS guidelines or sound programming practices. Continue with the following steps to determine which is the case.

6. Test whether PC-Interface presence causes the error, as follows:

   a. Restore the file CONFIG.XYX to its original contents by copying the file you saved back onto CONFIG.SYS.

   b. Restart the system noting that this time banners should appear for the device drivers that are installed as part of PC-Interface. However, no banner should appear that says that PC-Interface has been initialized

   c. Do not run the program, PCIINIT from the AUTOEXEC.BAT file or manually, since this would activate PC-Interface.

   d. PC-Interface is now present in the system memory but is passive. It has no effect on the system other than taking up a certain amount of memory.

   e. Now perform the sequence that produced the error.

   Is the error still present?

   Yes: the DOS program that fails in this case (but not when PC-Interface is NOT resident in memory) is written incorrectly but may work without PC-Interface. Some application programs use DOS system calls the PC-Interface does not support. Contact your PC-Interface vendor to determine whether your application program is known to cause PC-Interface problems.

   No: If you have not made any mistakes, you have found a PC-Interface problem and should contact your PC-Interface vendor for assistance. Please be prepared to give the vendor all relevant information concerning the problem.

Appendix A. PC-Interface Messages

This appendix lists all PC-Interface host messages that the adminstrator might see. During execution of PC-Interface, the error logger **(/usr/pci/errlogger)** mails messages to root on the host system and also displays them on the host system console.

Messages can also be generated during installation or deinstallation of the PC-Interface host software. These messages are not handled by the **errlogger** program, but are displayed on the host system console.

Messages that can appear during execution of PC-Interface are listed first below, followed by the installation and deinstallation messages. Each message is accompanied by an explanation of the cause of the message and a recommended response. In the messages, italics denote generic information for which the system substitutes actual values. For example, *device name* in a message is replaced by the name of an actual device, and *amount* and *n* are replaced by numbers.

The abbreviation **errno** stands for UNIX system error number. When an error number appears in a PC-Interface message, it means there was an error condition in the underlying UNIX system. To find out the cause of the problem, consult the list that correlates the error number with a description of the cause. On many systems this list is on-line in the file **/usr/include/sys/errno.h.** If your system does not have this file, consult your host system documentation.

PC-Interface Execution-Time Messages

These are messages that can be seen during execution of PC-Interface. For other messages appearing while using PC-Interface, refer to the *OfficeLAN Administrator's Guide.*

- **Can't exec DOS server 'usr/pci/pcidossvr.eth': *errno***
- **Can't exec DOS server '/usr/pci/pcidossvr.232': *errno***

  Explanation: The **pciconsvr** could not start the **pcidossvr** process.

  User Response: Check that the file **/usr/pci/pcidossvr.eth** or **/usr/pci/pcidossvr.232** exists and has execute permission for all UNIX users. The personal computer user should try logging in to PC-Interface again.

- **Can't exec /usr/pci/pcidosout.232: *errno***
- **Can't exec /usr/pci/pcidosout.eth: *errno***

  Explanation: This is a terminal emulation error. The **pcidossvr** could not start the **pcidosout** process.

  User Response: Check that the file **/usr/pci/pcidosout.eth** or **/usr/pci/pcidosout.232** exists and has execute permission for all UNIX users. The personal computer user should retry terminal emulation.

- **Can't open "Ethernet n" errno: *errno***

  Explanation: The PC-Interface connection server **(pciconsvr)** cannot access Ethernet probably because an errant PC-Interface server is tying up an Ethernet port.

  User Response: Shut down PC-Interface and restart it. If the problem persists, check the reported error number for the specific cause of the problem and take appropriate action to restore UNIX resources.

- **Can't open network — Bye**

  Explanation: This is a PC-Interface initialization problem. The **pciconsrv** tried to open the network for normal PC-Interface communication and failed. The PC-Interface files are probably corrupted.

  User Response: Check that all PC-Interface files are installed properly, are accessible, and are not corrupted. If necessary, reinstall the PC-Interface host software. Restart PC-Interface by invoking **/usr/pci/pcistart.**

- **Connection server can't open network**

  Explanation: The PC-Interface connection server **(/usr/pci/pciconsvr.eth)** could not open a network port. The most likely cause is that the port is already open because the connection server is already running. Only one connection server process can be running on a host.

  User Response: None required. Do not try to execute the connection server when it is already running.

- **do connect:dev *device name:* Invalid serial number: serial n**

  Explanation: A user tried to log into the host via PC-Interface using a diskette with an invalid serial number. The invalid serial number is shown in the message. When this condition occurs, the host aborts the login attempt. The most likely cause is that the user attempted to illegally circumvent the PC-Interface copy-protection mechanism.

  User Response: No action required on host side. Be sure that users understand the proper use of PC-Interface diskettes.

- **Error getting current working directory.**

  Explanation: PC-Interface log in over RS-232 failed because the UNIX system call **getpwuid** or **getuid** failed.

  User Response: Verify that the user has a valid UNIX account, including a proper entry in **/etc/passwd** and a home directory. The user should retry logging in.

- **error reading conninfo pipe: *errno***

  Explanation: This is a terminal emulation error. The communication between the **pcidossvr** and the **pcidosout** process failed.

  User Response: Check the reported error number for the specific cause of the problem and take appropriate action to restore UNIX resources. Retry emulation.

- **Error *errno* on get tty**

  Explanation: The initialization of the terminal port for terminal emulation over RS-232 failed.

  User Response: Refer to the reported error number for a precise description of the problem. Take appropriate action to restore UNIX system resources.

- **Error *errno* on NIGETA; Bye!**

  Explanation: This is an Ethernet driver error message. The network interface get-address (NIGETA) routine, which determines the host Ethernet address, failed.

  User Response: Refer to the reported error number and take appropriate action.

- **Error *errno* on NISETA; Bye!**

  Explanation: Initialization of the Ethernet device driver failed. The network interface set-parameters (NISETA) routine, which establishes parameters for network communication, failed.

  User Response: Refer to the reported error number and take appropriate action.

- **Error** errno **on STDIN set tty**

  Explanation: An operation on standard input failed.

  User Response: Refer to the reported error number and take appropriate action.

- **IO ERROR ON PIPE**

  Explanation: This is a terminal emulation error. An error condition on the pipe between the **pcidossvr** and the **pcidosout** process has occurred.

  User Response: Retry terminal emulation.

- **memory: Can't get *amount* bytes**

  Explanation: The PC-Interface server process cannot allocate memory required for its tables. The UNIX system call **malloc** failed.

  User Response: The system may be in an instable state and should be rebooted.

- **memory: Can't resize to *amount***

  Explanation: The PC-Interface server process cannot allocate memory required for its tables. The UNIX sysem call **malloc** failed.

  User Response: The system may be in an unstable state and should be rebooted.

- **NISETA Error: *errno***

  Explanation: The Ethernet device driver needs system resources that aren't available.

  User Response: Refer to the reported error number for the cause of the condition and take appropriate action to restore system resources.

- **PIPE Read error; *errno***

  Explanation: This is a terminal emulation error. There was an I/O error on the pipe between the **pcidossvr** and the **pcidosout** process.

  User Response: Refer to the reported error number and take appropriate action to restore system resources. Retry terminal emulation.

- **PIPE Write error: *errno;* Bye!**

  Explanation: This is a terminal emulation error. There was an I/O error on the pipe between the **pcidossvr** and the **pcidosout** process.

  User Response: Refer to the reported error number and take appropriate action to restore system resources. Retry terminal emulation.

- **Probe timeout - disconnect dev *names,* pid *n***

  Explanation: A personal computer has disconnected from the host without logging out. The host has killed PC-Interface processes associated with this personal computer.

  User Response: None required on the host. This condition is fatal for the personal computer. To reestablish a host connection, the user must log in again.

- **PTY EOF; Bye**

  Explanation: This is a terminal emulation error. The **pcidosout** process encountered an error trying to read from the pty.

  User Response: Retry terminal emulation.

- **PTY Read error: *errno***

  Explanation: This is a terminal emulation error. The **pcidosout** process encountered an error trying to read from the pty.

  User Response: Check the reported error number for the cause of the problem and take appropriate action to restore system resources. Retry terminal emulation.

- **PTY Write error: *errno;* Bye!**

  Explanation: This is a terminal emulation error. The terminal emulator tried to write characters to the pseudo-terminal (pty) driver and failed.

  User Response: Refer to the reported error number for the cause of the problem and take appropriate action to restore system resources. Retry terminal emulation.

- **Too many reXMITS**

  Explanation: This is a terminal emulation error. The **pcidosout** process tried to transmit data to the personal computer, but the personal computer did not acknowledge the transmission. The terminal emulation process is therefore terminated.

  User Response: Retry terminal emulation.

- **TTY Write error: *errno***

  Explanation: This is a terminal emulation error. The terminal emulator tried to write to a terminal device and failed.

  User Response: Refer to the reported error number for the cause of the problem and take appropriate action to restore system resources.

Installation and Deinstallation Diagnostics

The following messages may appear during the installation or deinstallation process.

- **mkdir failed**

  Explanation: The install script failed to create the following directories: **/usr/pci, /usr/pci/bin, /usr/src/uts/m68k/M68020/VME131cf/pci, /usr/spool/pcilog,**

  User Response: Make sure that the /usr file system is mounted and that it has write permission.

- **edit of /usr/scr/uts/m68k/Kmodfiles failed**

  Explanation: The install script for the 8000 system was not able to edit the file **/usr/pci/uts/m68k/Kmodfiles.**

  User Response: Make sure that the file exists and that it has write permission.

- **---- You must be logged in as root! ----**

  Explanation: You must be logged in as root in order to deinstall the PC-Interface software.

  User Response: Log in as root and repeat the deinstallation procedure.